

**PATENT APPLICATION**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of

Docket No: Q102939

Roger MAITLAND, et al.

Appln. No.: 10/762,364

Group Art Unit: 2434

Confirmation No.: 4471

Examiner: TRAN, ELLEN C

Filed: January 23, 2004

For: METHODS AND APPARATUS FOR PARALLEL IMPLEMENTATIONS OF TABLE  
LOOK-UPS AND CIPHERING

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**

**MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 41.37, Appellant submits the following:

**Table of Contents**

I. REAL PARTY IN INTEREST .....	.2
II. RELATED APPEALS AND INTERFERENCES .....	.3
III. STATUS OF CLAIMS .....	.4
IV. STATUS OF AMENDMENTS .....	.5
V. SUMMARY OF THE CLAIMED SUBJECT MATTER .....	.6
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL .....	.9
VII. ARGUMENT .....	10
CLAIMS APPENDIX .....	15
EVIDENCE APPENDIX: .....	.41
RELATED PROCEEDINGS APPENDIX .....	.42

**L. REAL PARTY IN INTEREST**

The real party in interest is Alcatel Lucent, the assignee.

**II. RELATED APPEALS AND INTERFERENCES**

There are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-79 are pending.

Claims 74-79 are rejected under 35 USC 103(a) as unpatentable over Kim et al (WO 03/050784) in view of Luyster (USP 6,751,319).

Claims 1, 2, 5, 6, 11-13, 16, 21-28, 30, 31, 33-42, 44, 45, 47-73 and 77-79 are rejected under 35 USC 103(a) as unpatentable over Kim et al in view of Luyster, and further in view of 3GPP TS 35.202 v3.1.1 Release 1999 (3GPP).

Claims 3, 4, 7-10, 14, 15, 17-20, 29, 32, 34, 43 and 46 are rejected under 35 USC 103(a) as unpatentable over Kim et al in view of Luyster and 3GPP, and further in view of Weybrew et al (USP 6,931,511).

All claims are appealed.

**IV. STATUS OF AMENDMENTS**

There are no amendments filed after the final Office mailed March 4, 2009.

#### **V. SUMMARY OF THE CLAIMED SUBJECT MATTER**

The following is a description of the claimed subject matter with reference to each of independent claims 1, 12, 21, 35, 49-51, 55, 64, and 73-76.

The context of the invention is implementation of an encryption process which involves the use of lookup tables. Fig. 1 shows a block cipher operation wherein input data 140 is subjected to an Exclusive-OR operation at 150 to obtain ciphered data. The ciphering input to 150 comes from a block cipher 10 and is a function of a key 120 and an input 110. The present invention takes place within the cipher block 100, and is a more efficient method and apparatus for generating the output 130. In the example given in the specification, the ciphering algorithm to be implemented is a Kasumi algorithm. One of the parts of the Kasumi Algorithm is an S7 function explained by the formulas of Fig. 3, where it can be seen that for each 7-bit input  $X = x_6x_5x_4x_3x_2x_1x_0$  there is a 7-bit output  $Y = y_6y_5y_4y_3y_2y_1y_0$  in which each bit of Y is a different logical function of the bits of X.

There are  $27 = 128$  possible different permutations for a 7-bit X value, and therefore  $27 = 128$  possible different values of Y. These are designated generally at 520 in Fig. 6. The 128 different values of Y are stored in a look-up table structure, but the look-up table is divided into four different tables 540, each containing 32 entries. While the look-up tables store Y values, it should be kept in mind that the labeling in Fig. 6 is not the Y values. Each different Y value will correspond to a different one of 128 possible permutations of bit values for X. Thus, in the notation of Fig. 6, “S7(0000001)” represents the S7 function of an X value of 0000001, it designates the 7-bit Y value that results from an X value of 0000001 according to the S7 function shown formulaically in Fig. 3.

Note that the entries in the top look-up table 540 all begin with the same x-value 00, the next table 540 contains entries all beginning with the most significant bits 01, the third table 10 and the fourth table 10. Within each table, there are 32 different values corresponding to 32

different permutations of the least significant five bits in combination with the particular pair of most significant bits.

The selection process is that, for each look-up table 540, the five least significant bits of the X input are used to select one of the 32 possible Y-values at step 581 in Fig. 6, thereby resulting in four different Y-value outputs 506. In the next two selection steps 582 and 583 in Fig. 6, the value of the two most significant bits of the current X value is then used to select one of the four Y-value outputs.

Finally, while Fig. 6 illustrates the process for a given value of X, it is noted that this process takes place in parallel for plural  $X_i$  inputs, as noted in the last three lines of 0057.

While the flow of the process is shown in Figs. 5 and 6 to implement the formulas in Fig. 3, the apparatus for doing so is illustrated in Fig. 19A. The look-up tables 540 in Fig. 6 are represented by memory 1810 in Fig. 19A, responding to a plurality of X inputs 1840 to produce a plurality of outputs, and the processor 1820 then uses the most significant bits to select one of the outputs for each X input, to produce outputs 1830.

Thus, in the language of claim 1, for plural inputs  $X_i$ , each input defined by a first set of bits (the five least significant bits) and a second set of bits (the two most significant bits), looking up one element from each look-up table 540 using the first set of bits to obtain a set of corresponding outputs 591-594, and then selecting a corresponding output from that set using the second set of bits.

In claim 12, the memory is shown at 1810, processor at 1820, and the functions performed by the processor 1820 being as described above for claim 1 and as described at paragraphs 0137 and 0138.

Claim 21 is directed to an alternative implementation shown in Figs. 11 and 20 and described beginning in paragraph 0142, where the plurality of bits making up each of N inputs are separated into subsets, Fig. 20 showing two subsets 2004 and 2006 for each of the N inputs. Each subset of input bits is used to access a look-up table, so that the N subsets 204 yield N

Group 1 outputs 2012 and the N subsets 2006 yield N Group M outputs 2014. respective ones of the Group 1 and Group 2 outputs are then combined at 2016 to obtain N outputs 2018.

Claim 35 is directed to an apparatus for performing the method of claim 21, and this is the same apparatus as in Fig. 19A except that the memory 1810 is organized differently and the processor 1820 follows a different process.

Claim 49 is directed to a computer readable medium carrying code for implementing the method of claim 1.

Claim 50 is directed to a computer readable medium carrying code for implementing the method of claim 21.

Claim 51 is directed to the method shown in Fig. 2o and described beginning at paragraph 0142.

Claim 55 is directed to a ciphering apparatus employing the method and apparatus of claims 1 and 12, but broadly directed to looking up in plural look-up tables in parallel for each of a plurality of first inputs.

Claim 64 is directed to the apparatus of Fig. 19A, with the memory at 1810 and the processor at 1820.

Claim 73 is directed to a computer readable medium carrying code for implementing the method of claim 55.

Claim 74 is directed to a method corresponding to the apparatus of claim 55.

Claim 75 is directed to an apparatus (Fig. 19A) for performing a method similar to claim 55.

Claim 76 is directed to a computer readable medium carrying code for practicing the method of claim 75.

**VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

The grounds of rejection to be reviewed on appeal are:

1. Whether claims 74-79 are unpatentable over Kim et al (WO 03/050784) in view of Luyster (USP 6,751,319).
2. Whether claims 1, 2, 5, 6, 11-13, 16, 21-28, 30, 31, 33-42, 44, 45, 47-73 and 77-79 are unpatentable over Kim et al in view of Luyster and further in view of 3GPP TS 35.202 v3.1.1 Release 1999 (3GPP).
3. Whether claims 3, 4, 7-10, 14, 15, 17-20, 29, 32, 34, 43 and 46 are unpatentable over Kim et al in view of Luyster and 3GPP, and further in view of Weybrew et al (USP 6,931,511).

## VII. ARGUMENT

### **Claims 74-79 Are Not Unpatentable Over Kim et al in view of Luyster**

Claims 74-76 are directed to the simultaneous use of different parts of a signal to address one or more lookup tables. The examiner acknowledges that Kim does not teach the use of lookup tables. The examiner cites Luyster for its teaching of look-up tables, but the novelty in claim 74 is not simply the use of a lookup table but the parallel use of different parts of a single input signal to address different lookup tables or different parts of a lookup table. This novelty is stated at lines 22-27 of page 2 of the present application as filed.

At page 6 of the final Office action, the examiner reads the upper 16 bits and lower 16 bits of the 32-bit input in Kim as the claimed plurality of inputs, but this is an unreasonable reading of the claim language and Kim et al. There is only one input in Kim et al, a 32-bit input. The upper and lower sets of bits are different parts of the same input, but are not a plurality of inputs as required in claims 74-76. The claim construction adopted by the examiner is based entirely (and improperly) on hindsight.

Further, the central point of claim 74 is that lookup tables or parts of lookup tables are accessed in parallel by different inputs, and it is clear from the description of Fig. 4 at page 9 of Kim that in the first pipeline section 310 the upper set of bits is processed while the lower set of bits is simply stored and then used to Exclusive-OR with the processed upper set to form an output of the stage 310. Then in the second stage 320 the lower set of bits is processed and the result is Exclusive-ORED with the output of the first stage. So the processing of the first and second sets of bits in Kim is not in parallel as is required by claims 74-76, but instead is in series, first the upper set, then the lower set. Thus, even if lookup tables as taught by Luyster were used in Kim, the result would not be the parallel use of the two different parts of the input to access lookup tables. Instead, the upper part might be used to access a lookup table to implement the processing of the upper bits, but the lower set of bits would not be used until the upper set was

done. Indeed, the second stage uses the results of processing the upper set of bits in order to generate the second stage output. This is the antithesis of “parallel” operation.

The result is that Kim et al uses two different parts of a *single* input and uses them *in series*, whereas the present invention uses *different inputs* and uses them *in parallel*. These are entirely different.

In Section II at page 3 of the Office action, the examiner responds to this distinguishing argument by asserting that the claims do not recite that the two accesses are simultaneous. While the examiner is correct that the term “simultaneously” is not used in claims 74-76, the examiner is incorrect in concluding that the concept of “simultaneous” can or should be ignored. One of ordinary skill in the art would understand that when two things are described as being done “in parallel” they are considered to be concurrent. Indeed, even the examiner concedes this, stating at page 6 of the Office action that “simultaneously is considered to be equivalent to in parallel.” The fact is that no one of skill in the art would consider an arrangement where one process is performed and then after that a second process is performed to be performing the two processes “in parallel.”

So if Kim were modified to incorporate lookup tables, it would still be the case that there would be no parallel use of plural different input signals as is required of claims 74-76.

**Claims 1, 2, 5, 6, 11-13, 16, 21-28, 30, 31, 33-42, 44, 45, 47-73 And 77-79 Are Not Unpatentable Over Kim et al In View of Luyster And Further In View of 3GPP TS 35.202 v3.1.1 Release 1999 (3GPP).**

Claim 1 describes each input as including a first set of bits used to access a lookup table, with first sets of bits from plural inputs accessing plural lookup tables to collectively obtain a set of outputs, and then one of the outputs from the set of outputs is selected using a second set from at least one of the inputs. The examiner refers to 3GPP as teaching the output of a first string used as an input to a second, but that is not what is required by the “selection” operation referred to in claim 1. Thus, the additional reference does not make up for the deficiencies already pointed out relative to Kim and Luyster.

Specifically, claim 1 requires plural inputs each having first and second sets of bits. The examiner reads the upper and lower sets of bits in Kim as the first and second sets of bits, but that means there is only one input shown in Kim. Not the plurality of inputs required by claim 1.

Further, if the upper 16 bits is the first set of bits and the lower 16 bits is the second set of bits, then in order to satisfy the requirements of claim 1 it would be necessary that plural sets of upper bits each be used to access respective lookup tables to collectively obtain a set of lookup table outputs, and then one of these outputs is selected using a bit from a lower set of bits.

Luyster teaches lookup tables, but does not teach the other subject matter missing from Kim, e.g., the plural inputs or the accessing of plural lookup tables in parallel by the first sets of the plural inputs, or the selection of one of the lookup table outputs. So adopting the teaching of Luyster in Kim would result in a single lookup table accessed by the single first set shown in Kim.

The examiner relies on 3GPP to teach this final selection, but there are two problems with this. First, there are no plural outputs to select from in Kim. Second, the examiner notes that 3GPP teaches that the output from the first bit string are used as inputs to the second string.” But this is not what is claimed. Claim 1 does not state that the output from one string is used as an input to a second string, but rather that a bit from the second set is used to select from amongst plural lookup table outputs. This is simply not shown in 3GPP.

Claims 12, 49, 51 and 77-79 recite the same distinctive features and distinguish over the prior art for the same reasons.

Regarding claim 21, that claim recites a plurality of inputs and the selection of a subset of bits from each input to use to access a respective lookup table, and then the combining of the lookup table outputs to obtain at least one bit. Kim does not teach the plurality of inputs, but according to the examiner a single input having first and second sets of bits. And again, the examiner cites 3GPP as teaching the use of the output of one bit string as an input to a second string, but this is not what is explicitly recited in claim 21, which is the combination of lookup table outputs (not using an output of one as an input to another).

Claims 35, 50, 51 and 59 distinguish over the cited art for the same reasons.

Claims 55, 64 and 73 all recite that for each of plural first inputs and in parallel with other first inputs, accessing a lookup table using the input. As discussed above in the context of claim 74, Kim shows only one input, and even if one looks at the upper and lower sets of bits as separate inputs, they do not access lookup tables in parallel, but rather first one set and then the other set is processed. The secondary references do not make up for this deficiency.

In the final Office action, the examiner again refers to the upper and lower bits as first and second sets of bits, but the examiner has missed the point. Claim 1 requires that there be plural inputs, with each input defined by first and second sets of bits. The upper and lower sets of bits in Kim et al are different parts of the same input, and may therefore correspond to the claimed first and second sets of bits. But claim 1 recites that the method is responsive to a plurality of inputs, and for each input and in parallel with other inputs, there are two steps performed. First, the first set of bits of the plurality of inputs are used to access plural lookup tables with the outputs from the lookup tables collectively forming a set of outputs. Second, the second set of bits of each input is used to select one output from the set of outputs. These two steps are performed for each input, and in parallel with other inputs.

If the upper and lower bits in Kim et al are considered to be the claimed first and second sets of bits, then the upper and lower bits together form an input. That is fine, but claim 1 would require that the upper bits from plural inputs access respective different lookup tables, and that the lower bits of each input signal be used to select from amongst the outputs of the plural lookup tables. This simply does not happen in Kim et al, nor would it happen if Kim et al's process were implemented using lookup tables, nor would it happen if "the output from the first bit string are used for inputs to the second string" as the examiner proposes at page 9 to adopt from 3GPP.

In Section IV at page 4, the examiner dismisses these arguments on multiple unsupportable grounds. First, the examiner argues that claim 1 does not require multiple lookup tables. The examiner is ignoring the clear language of the second subparagraph of claim 1 which reads:

Appeal Brief Under 37 C.F.R. § 4.37  
USSN 10/762,364

looking-up one of a plurality of elements of each of a plurality of look-up tables using the first set of bits that define the input to obtain an output, the outputs from each of the plurality of look-up tables collectively comprising a set of corresponding outputs;

The examiner further argues that the Kasumi algorithm uses plural inputs, but this does not lead to the use of plural lookup tables in the particular manner recited in claim 1.

In Section V at page 5, the examiner purports to dismiss applicants arguments, but the dismissal is not warranted. The examiner somehow concludes that because the Kasumi algorithm uses multiple rounds of processing the selection operation of claim 1 is inherent. There is simply no support for this. Using the output of one round as the input to the next is not what is recited in claim 1. Claim 1 describes the use of the first parts of plural inputs to access plural lookup tables to generate a plurality of lookup table outputs, and then the second part of each input being used to select one of the set of outputs. This is not the use of one stage output as an input to the next stage.

The remaining independent claims all distinguish on the same basis as claim 1, and all dependent claims distinguish by virtue of their dependency. Accordingly, reversal of the examiner is requested.

**Claims 3, 4, 7-10, 14, 15, 17-20, 29, 32, 34, 43 And 46 Are Not Unpatentable Over Kim et al In View of Luyster And 3GPP, And Further In View of Weybrew et al**

These claims are all patentable due to dependence on patentable parent claims as discussed above.

Respectfully submitted,

/DJCushing/

David J. Cushing  
Registration No. 28,703

SUGHRUE MION, PLLC  
Telephone: (202) 293-7060  
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Date: April 23, 2010

**CLAIMS APPENDIX**

CLAIMS 1-79 ON APPEAL:

1. A method comprising, responsive to a plurality of inputs, each input being defined by a first set of bits and a second set of at least one bit, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs:

looking-up one of a plurality of elements of each of a plurality of look-up tables using the first set of bits that define the input to obtain an output, the outputs from each of the plurality of look-up tables collectively comprising a set of corresponding outputs; and

selecting a corresponding output from the set of corresponding outputs using the second set of at least one bit that defines the input.

2. A method according to claim 1 wherein the plurality of elements of each look-up table collectively comprise a combined table of elements each having a pre-determined value obtained using an S7 function.

3. A method according to claim 1 wherein for each look-up table, the plurality of elements of the look-up table and the plurality of inputs are loaded as vectors and the looking-up comprises, for each of the inputs, selecting one of the plurality of elements of the look-up table using the first set of bits that define the input.

4. A method according to claim 3 comprising using a vperm (vector permutation) instruction for selecting one of the plurality of elements of the look-up table using the first set of bits that define the input.

5. A method according to claim 1 wherein, for each of the plurality of inputs, the second set of at least one bit that defines the input comprises one bit and the set of corresponding outputs comprises two corresponding outputs, and wherein for each of the plurality of inputs the selecting comprises:

selecting one of the two outputs using the one bit of the at least one bit that defines the input.

6. A method according to claim 1 wherein, for each of the plurality of inputs, the second set of at least one bit that defines the input comprises at least two bits, and wherein for each of the plurality of inputs the, selecting comprises:

successively performing a selection on a remaining number of corresponding outputs of the set of corresponding outputs for each bit of the at least two bits, the number of corresponding outputs remaining being equal to all of the corresponding outputs of the set of corresponding outputs a first time the selection is performed, the selection being replacing the remaining number of corresponding outputs with a selection of half of the remaining number of outputs using a respective bit of the at least two bits, the selection of half of the remaining number of outputs being the number of remaining outputs for the next time the selection is performed.

7. A method according to claim 6 wherein, for each time the selection on a remaining number of corresponding outputs is performed, the remaining number of corresponding outputs comprises at least one set of two remaining corresponding outputs and the selection of half of the remaining number of outputs comprises, for each set of two corresponding outputs of the at least one set of two remaining corresponding outputs: replicating the respective bit into a plurality of replicated bits; and using a vector instruction, selecting one of the two remaining corresponding outputs depending on the plurality of replicated bits.

8. A method according to claim 7 wherein the vector instruction is a vsel (vector select instruction).

9. A method according to claim 2 wherein, for each input, the first set of bits that define the input comprises five bits, the second set of bits that define the input comprises two bits and the look-up tables comprise four look-up tables, wherein for each of the four look-up tables the plurality of inputs and the plurality of elements of the look-up table are loaded as vectors and the looking-up comprises for each of the inputs selecting one of the plurality of elements of the look-up table using the first set of bits that define the input.

10. A method according to claim 2 wherein, for each input, the first set of bits that define the input comprises four bits, the second set of bits that define the input comprises three bits and the look-up tables comprise eight look-up tables, and wherein for each of the eight look-up tables the plurality of inputs and the plurality of elements of the lookup table are loaded as vectors and for each of the inputs the looking-up comprises selecting one of the plurality of elements of the look-up table using the first set of bits that define the input.

11. A method according to claim 2 applied in ciphering data in a Kasumi implementation.

12. An apparatus comprising:  
a memory adapted to store a plurality of elements of each of a plurality of look-up tables; and  
a processor for:

receiving a plurality of inputs, each input being defined by a first set of bits and a second set of at least one bit, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs:

looking up one of a plurality of elements of each of a plurality of look-up tables using the first set of bits that define the input to obtain an output, the outputs from the plurality of look-up tables collectively comprising a set of corresponding outputs; and

selecting a corresponding output from the set of corresponding outputs using the second set of at least one bit that define the input.

13. An apparatus according to claim 12 wherein the plurality of elements of each look-up table collectively comprise a combined table of elements each having a pre-determined value obtained using an S7 function.

14. An apparatus according to claim 12 wherein, for each look-up table, the plurality of elements of the look-up table and the plurality of inputs are loaded as vectors and for each of the inputs the processor is further adapted to select one of the plurality of elements of the look-up table using the first set of bits that define the input.

15. An apparatus according to claim 14 wherein the processor comprises a co-processor having a vperm (vector permutation) instruction, the processor being adapted to use the vperm instruction for the selecting one of the plurality of elements of the look-up table using the first set of bits that define the input.

16. An apparatus according to claim 12 wherein, for each of the plurality of inputs, the second set of at least one bit that defines the input comprises at least two bits, and wherein for each of the plurality of inputs in selecting the corresponding output from the set of corresponding outputs the processor is adapted to:

successively perform a selection on a remaining number of corresponding outputs of the set of corresponding outputs for each bit of the at least two bits, the number of corresponding outputs remaining being equal to all of the corresponding outputs of the set of corresponding outputs a first time the selection is performed, the selection being replacing the remaining number of corresponding outputs with a selection of half of the remaining number of outputs using a respective bit of the at least two bits, the selection of half of the remaining number of outputs being the number of remaining outputs for the next time the selection is performed.

17. An apparatus according to claim 16 wherein, for each time the selection on a remaining number of corresponding outputs is performed, the remaining number of corresponding outputs comprises at least one set of two remaining corresponding outputs and the selection of half of the remaining number of outputs comprises, for each set of two corresponding outputs of the at least one set of two remaining corresponding outputs the processor being adapted to:

replicate the respective bit into a plurality of replicated bits; and  
using a vector instruction, select one of the two remaining corresponding outputs depending on the plurality of replicated bits.

18. An apparatus according to claim 17 wherein the processor comprises a co-processor having a vsel (vector select instruction), the vsel instruction being the vector instruction.

19. An apparatus according to claim 13 wherein, for each input, the first set of bits that define the input comprises five bits, the second set of bits that define the input comprises two bits and the look-up tables comprise four look-up tables, wherein for each of the four look-up tables the plurality of inputs and the plurality of elements of the look-up table are loaded as vectors and for each of the inputs the processor is adapted to select one of the plurality of elements of the look-up table using the first set of bits that define the input.

20. An apparatus according to claim 13 wherein, for each input, the first set of bits that define the input comprises four bits, the second set of bits that define the input comprises three bits and the look-up tables comprise eight look-up tables, and wherein for each of the eight look-up tables the plurality of inputs and the plurality of elements of the look-up table are loaded as vectors and for each of the inputs the processor is adapted to select one of the plurality of elements of the look-up table using the first set of bits that define the input.

21. A method comprising:  
responsive to a plurality of inputs each defined by a first plurality of bits, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs and for each of a plurality of look-up tables each having a plurality of elements:

selecting a respective subset of bits of the first plurality of bits that define the input, the bits of the respective subset of bits comprising fewer bits than the first plurality of bits of the input; and

looking-up an element of the plurality of elements of the look-up table using the subset of bits to obtain an output; and

combining the outputs obtained from the plurality of look-up tables to obtain at least one bit.

22. A method according to claim 21 wherein, for each input of the plurality of inputs, the outputs obtained from the plurality of look-up tables each comprise a second plurality of bits, the second plurality of bits comprising fewer bits than the first plurality of bits of the input.

23. A method according to claim 22 wherein, for each input of the plurality of inputs, the at least one bit comprises a third plurality of bits, the third plurality of bits comprising the same number of bits as the first plurality of bits of the input.

24. A method according to claim 21 wherein, for at least one look-up table of the plurality of look-up tables, for each input the selecting comprises manipulating at least one of the plurality of bits that define the input using at least one of a bit rotation instruction and a bit shifting instruction.

25. A method according to claim 24 wherein, for each of the at least one look-up table, for each input the manipulating at least one of the first plurality of bits comprises ordering the respective subset of bits of the input as least significant bits.

26. A method according to claim 23 wherein each element of the plurality of elements of each look-up table has a pre-determined value.

27. A method according to claim 26 wherein, for each input of the plurality of inputs the first plurality of bits and the third plurality of bits each comprise 9 bits, the pre-determined value of each of the plurality of elements of each of the plurality of look-up tables is obtained from a partial evaluation of an S9 function.

28. A method according to claim 27 wherein, for each look-up table of the plurality of look-up tables, the pre-determined value of each of the plurality of elements of the look-up table is a function of a number being definable by a bit sequence of one of 4 and 5 bits.

29. A method according to claim 28 wherein, for each input of the plurality of inputs, for each look-up table the respective subset of bits of the first plurality of bits that define the input comprises one of 4 and 5 bits and the look-up table is looked-up using a vperm (vector permutation) instruction.

30. A method according to claim 27 wherein, for each input of the plurality of inputs, the combining comprises performing a plurality of exclusive-OR operations on the outputs obtained from the plurality of look-up tables for the input.

31. A method according to claim 30 wherein for each input of the plurality of inputs, the combining comprises manipulating the second plurality of bits of at least one output of the outputs obtained from the plurality of look-up tables for the input using one of a bit shifting instruction and a bit rotation instruction.

32. A method according to claim 31 wherein the bit shifting instruction comprises one of a vector shift right byte instruction and a vector shift left byte instruction and the bit rotation instruction comprises one of a vector rotate left byte instruction and a vector rotate right byte instruction.

33. A method according to claim 30 wherein, for each input of the plurality of inputs, the combining comprises:

for a first output of the outputs obtained from the plurality of look-up tables for the input, manipulating the second plurality of bits of the first output using one of a bit rotation instruction and a bit shifting instruction; and

for a second output of the outputs obtained from the plurality of look-up tables for the input, performing one of the plurality of exclusive-OR operations on the second output and the first output to obtain a third output having a fourth plurality of bits.

34. A method according to claim 30 wherein, for each input, the bits of the second plurality of bits of each respective subset of bits of the first plurality of bits of the input have a pre-determined order and are each used for obtaining a respective one of the third plurality of bits, the outputs obtained from the look-up tables collectively comprising at least one group of outputs each having at least two outputs of the outputs obtained from the look-up tables,

for each group of outputs of the at least one group of outputs the at least two outputs in the group of outputs having bits used for determining a common subset of bits of the third plurality of bits, the combining comprising:

for each group of outputs of the at least one group of outputs, combining the at least two outputs of the group of outputs using at least one of the plurality of exclusive-OR operations.

35. An apparatus comprising:

a memory adapted to store a plurality of elements of each of a plurality of look-up tables; and

a processor responsive to a plurality of inputs each defined by a first plurality of bits, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs and for each look-up table of the plurality of look-up tables, for:

selecting a respective subset of bits of the first plurality of bits that define the input, the bits of the respective subset of bits comprising fewer bits than the first plurality of bits of the input;

looking up an element of the plurality of elements of the look-up table using the subset of bits to obtain an output; and

combining the outputs obtained from the plurality of look-up tables to obtain at least one bit.

36. An apparatus according to claim 35 wherein, for each input of the plurality of inputs, the outputs obtained from the plurality of look-up tables each comprise a second plurality of bits, the second plurality of bits comprising fewer bits than the first plurality of bits of the input.

37. An apparatus according to claim 36 wherein, for each input of the plurality of inputs, the at least one bit comprises a third plurality of bits, the third plurality of bits comprising the same number of bits as the first plurality of bits of the input.

38. An apparatus according to claim 35 wherein, for at least one look-up table of the plurality of look-up tables, and for each input, the processor is adapted to manipulate at least one of the first plurality of bits that define the input using at least one of a bit rotation instruction and a bit shifting instruction.

39. An apparatus according to claim 38 wherin, for each of the at least one look-up table:

for each input the processor is adapted to manipulate the at least one of the first plurality of bits by ordering the respective subset of bits of the input as least significant bits.

40. An apparatus according to claim 37 wherein each element of the plurality of elements of each look-up table has a pre-determined value.

41. An apparatus according to claim 40 wherein, for each input of the plurality of inputs, the first plurality of bits and the third plurality of bits each comprise 9 bits, the pre-determined value of each of the plurality of elements of each of the plurality of look-up tables is obtained from a partial evaluation of an S9 function.

42. An apparatus according to claim 41 wherein, for each look-up table of the plurality of look-up tables, the pre-determined value of each of the plurality of elements of the look-up table is a function of a number being definable by a bit sequence of one of 4 and 5 bits.

43. An apparatus according to claim 42 wherein, for each input of the plurality of inputs, for each look-up table the respective subset of bits of the first plurality of bits that define

the input comprises one of 4 and 5 bits, the processor being adapted to look-up the look-up table using a vperm (vector permutation) instruction.

44. An apparatus according to claim 41 wherein, for each input of the plurality of inputs, the processor is adapted to perform a plurality of exclusive-OR operations on the outputs obtained from the plurality of look-up tables for the input.

45. An apparatus according to claim 44 wherein, for each input of the plurality of inputs, the processor is adapted to manipulate the second plurality of bits of at least one output of the outputs using one of a bit shifting instruction and bit rotation instruction.

46. A method according to claim 45 wherein the bit shifting instruction comprises one of a vector shift right byte instruction and a vector shift left byte instruction and the bit rotation instruction comprises one of a vector rotate left byte instruction and a vector rotate right byte instruction.

47. An apparatus according to claim 44 wherein, for each input of the plurality of inputs, the processor is adapted to:

for a first output of the outputs obtained from the plurality of look-up tables for the input, manipulate the second plurality of bits of the first output using one of a bit rotation instruction and a bit shifting instruction; and

for a second output of the outputs obtained from the plurality of look-up tables for the input, perform one of the plurality of exclusive-OR operations on the second output and the first output to obtain a third output having a fourth plurality of bits.

48. An apparatus according to claim 44 wherein, for each input, the bits of the second plurality of bits of each respective subset of bits of the first plurality of bits of the input have a pre-determined order and are each used for obtaining a respective one of the third plurality of bits, the outputs obtained from the look-up tables collectively comprising at least one group of outputs each having at least two outputs of the outputs obtained from the look-up tables, for each group of outputs of the at least one group of outputs the at least two outputs in the group of outputs having bits used for determining a common subset of bits of the third plurality of bits, the processor being adapted to:

for each group of outputs of the at least one group of outputs, combine the at least two outputs of the group of outputs using at least one of the plurality of exclusive-OR operations.

49. An article of manufacture comprising:  
a computer readable medium having computer readable program code means embodied therein, the computer readable code means in said article of manufacture comprising, responsive to a plurality of inputs, each input being defined by a first set of bits and a second set of at least one bit, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs;

computer readable code means for looking-up one of a plurality of elements of each of a plurality of look-up tables using the first set of bits that define the input to obtain an output, the output from each of the plurality of look-up tables collectively comprising a set of corresponding outputs; and

computer readable code means for selecting a corresponding output from the set of corresponding outputs using the second set of at least one bit that defines the input.

50. An article of manufacture comprising:

a computer readable medium having computer readable program code means embodied therein, the computer readable code means in said article of manufacture comprising, responsive to a plurality of inputs each defined by a first plurality of bits, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs:

computer readable code means for, for each of a plurality of look-up tables each having a plurality of elements:

selecting a respective subset of bits of the first plurality of bits that define the input, the bits of the respective subset of bits comprising fewer bits than the first plurality of bits of the input; and

looking-up an element of the plurality of elements of the look-up table using the subset of bits to obtain an output; and

computer readable code means for combining the outputs obtained from each look-up table to obtain at least one bit.

51. A method comprising, responsive to  $N K_i n$ -bit inputs:  
performing bit reordering on the  $N K_i n$ -bit inputs to produce  $M$  parallel sets of outputs  
wherein  $N$  and  $K_i n$  are integers satisfying  $N, K_i n \geq 2$ , an  $i$ th set of outputs of the  $M$  parallel sets  
of outputs containing  $N$  sets of bits  $L_{i,in}$  bits in length with  $i$  and  $L_{i,in}$  being integers satisfying  
 $i = 1$  to  $M$  and  $1 \leq L_{i,in} < K_i n$ , the  $i$ th set of outputs defining a respective subset of the  $K_i n$   
bits of the inputs;

for each parallel set of outputs, performing a parallel lookup table operation to generate a  
corresponding parallel set of outputs containing  $N$  outputs, each being associated with a  
respective one of the  $N K_i n$ -bit inputs and each being  $L_{i,out}$  bits in length,  $L_{i,out}$  being an integer  
satisfying  $L_{i,out} \geq 1$ ; and

for each of the  $N K_i n$ -bit inputs, generating a respective output by performing a bit  
combining operation on the outputs from the parallel look-up table operations associated with the  
input.

52. A method according to claim 51 wherein, for each of the  $N K_i n$ -bit inputs, the  
generating comprises performing a bit manipulation on the outputs of the parallel look-up table  
operations associated with the input.

53. A method according to claim 51 wherein the bit combining operations are  
implemented in parallel.

54. A method according to claim 51 wherein, for each of the  $N K_i n$ -bit inputs, the respective output generated comprises  $K_{out}$  bits,  $K_{out}$  being an integer satisfying  $K_{out} \geq 1$ , and wherein in performing the bit permutation/reordering on the  $N K_i n$ -bit inputs, the  $i$ th set of outputs defining the respective subset of the  $K_i$  bits of the inputs is selected such that the respective subset of the  $K_i$  bits effects only a defined maximum number  $P_i < K_{out}$  bits of the respective outputs wherein  $P_i$  is an integer.

55. A method of generating a plurality of outputs according to a ciphering algorithm which for each of the plurality of outputs operates on a respective input using a respective key, the ciphering algorithm comprising a plurality of rounds in which functions are evaluated, the method comprising, for at least one function of the functions of at least one of the plurality of rounds:

responsive to a plurality of first inputs each being associated with one of the respective inputs, for each first input and in parallel with other first inputs of the plurality of first inputs:

generating an output by looking up at least one look-up table using the input, each look-up table having a plurality of elements.

56. A method according to claim 55 wherein the ciphering algorithm is a Kasumi algorithm.

57. A method according to claim 55 wherein, for a function of a certain type of the at least one function, the at least one look-up table comprising a plurality of look-up tables and the output from each of the plurality of look-up tables collectively comprising a set of corresponding outputs, each first input of the plurality of first inputs being defined by a first set of bits and a second set of at least one bit, the method comprising for each first input of the plurality of first inputs and in parallel with the other first inputs of the plurality of first inputs:

selecting a corresponding output from the set of corresponding outputs using the second set of at least one bit that defines the input.

58. A method according to claim 57 wherein the ciphering algorithm is a Kasumi algorithm and the function of a certain type is an S7 function.

59. A method according to claim 55 wherein, for a function of a certain type of the at least one function, the at least one look-up table comprises a plurality of look-up tables and each first input of the plurality of first inputs is defined by a first plurality of bits, the method comprising:

for each first input of the plurality of first inputs and in parallel with the other first inputs of the plurality of first inputs, and for each of the plurality of look-up tables:

selecting a respective subset of bits of the first plurality of bits that define the first input, the bits of the respective subset of bits comprising fewer bits than the first

plurality of bits of the first input, the look-up table being looked up using the subset of bits to obtain the output; and

combining the outputs obtained from the plurality of look-up tables to obtain at least one bit.

60. A method according to claim 59 wherein the ciphering algorithm is a Kasumi algorithm and the function of a certain type is an S9 function.

61. A method according to claim 56 wherein the at least one round comprises the plurality of rounds and wherein for each round the at least one function comprises six S7 functions and six S9 functions, the method further comprising for each function of the plurality of functions other than the at least one function, and responsive to a plurality of second inputs each being associated with one of the respective inputs, and in parallel with other second inputs of the plurality of second inputs:

generating an output according to the function using the input.

62. A method according to claim 55 further comprising, for each output of the plurality of outputs and in parallel with other outputs of the plurality of outputs:

combining the output with input data to generate ciphered data.

63. A method according to claim 62 wherein the combining comprises performing an exclusive-OR operation.

64. An apparatus for generating a plurality of outputs according to a ciphering algorithm which for each of the plurality of outputs operates on a respective input using a respective key, the ciphering algorithm comprising a plurality of rounds in which functions are evaluated, the apparatus comprising:

a memory adapted to store a plurality of elements of each of at least one look-up table;  
and

a processor adapted to, for at least one function of the functions of at least one of the plurality of rounds, and responsive to a plurality of first inputs each being associated with one of the respective inputs, and for each first input and in parallel with other first inputs of the plurality of first inputs:

generate an output by looking up at least one look-up table using the input, each look-up table having a plurality of elements.

65. An apparatus according to claim 64 wherein the ciphering algorithm is a Kasumi algorithm.

66. An apparatus according to claim 64 wherein, for a function of a certain type of the at least one function, the at least one look-up table comprises a plurality of look-up tables

and the output from each of the plurality of look-up tables collectively comprising a set of corresponding outputs, each first input of the plurality of first inputs being defined by a first set of bits and a second set of at least one bit, the processor being further adapted to, for each first input of the plurality of first inputs and in parallel with the other first inputs of the plurality of first inputs:

select a corresponding output from the set of corresponding outputs using the second set of at least one bit that defines the input.

67. An apparatus according to claim 66 wherein the ciphering algorithm is a Kasumi algorithm and the function of a certain type is an S7 function.

68. An apparatus according to claim 64 wherein, for a function of a certain type of the at least one function, the at least one look-up table comprises a plurality of look-up tables and each first input of the plurality of first inputs is defined by a first plurality of bits, the processor being further adapted to, for each first input of the plurality of first inputs and in parallel with the other first inputs of the plurality of first inputs, and for each of the plurality of look-up tables:

select a respective subset of bits of the first plurality of bits that define the first input, the bits of the respective subset of bits comprising fewer bits than the first plurality of bits of the first input, the look-up table being looked up using the subset of bits to obtain the output;

and

combine the outputs obtained from the plurality of look-up tables to obtain at least one bit.

69. An apparatus according to claim 68 wherein the ciphering algorithm is a Kasumi algorithm and the function of a certain type is an S9 function.

70. An apparatus according to claim 65 wherein the at least one round comprises the plurality of rounds and wherein for each round the at least one function comprises six S7 functions and six S9 functions, the processor being further adapted to, for each function of the plurality of functions other than the at least one function, and responsive to a plurality of second inputs each being associated with one of the respective inputs, and in parallel with other second inputs of the plurality of second inputs:

generate an output according to the function using the input.

71. An apparatus according to claim 64 wherein the processor is further adapted to, for each output of the plurality of outputs and in parallel with other outputs of the plurality of outputs:

combine the output with input data to generate ciphered data.

72. An apparatus according to claim 71 wherein the processor is adapted to combine the output with the input data using an exclusive-OR operation.

73. An article of manufacture comprising:

a computer readable medium having computer readable program code means embodied therein for generating a plurality of outputs according to a ciphering algorithm which for each of the plurality of outputs operates on a respective input using a respective key, the ciphering algorithm comprising a plurality of rounds in which functions are evaluated, the computer readable code means in said article of manufacture comprising:

computer readable code means for, for at least one function of the functions of at least one of the plurality of rounds, and responsive to a plurality of first inputs each being associated with one of the respective inputs, for each first input and in parallel with other first inputs of the plurality of first inputs, generating an output by looking up at least one look-up table using the input, each look-up table having a plurality of elements.

74. A method comprising the step of, responsive to a plurality of inputs, each input being defined by at least one bit, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs, looking-up a look-up table having a plurality of elements using the at least one bit that define the input to obtain an output.

75. An apparatus comprising:

a memory adapted to store a plurality of elements of a look-up table; and

a processor adapted to, responsive to a plurality of inputs, each input being defined by at least one bit, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs look-up the look-up table using the at least one bit that define the input to obtain an output.

76. An article of manufacture comprising:

a computer readable medium having computer readable program code means embodied therein, the computer readable code means in said article of manufacture comprising:

computer readable code means for, responsive to a plurality of inputs, each input being defined by at least one bit, for each input of the plurality of inputs and in parallel with other inputs of the plurality of inputs, looking-up a look-up table having a plurality of elements using the at least one bit that define the input to obtain an output.

77. A method according to claim 74, wherein the look-up table outputs corresponding to the plurality of inputs comprise a set of outputs, and said method further comprises the step of selecting one of said outputs in response to at least one additional bit included in at least one of said plurality of inputs.

78. An apparatus according to claim 75, wherein the look-up table outputs corresponding to the plurality of inputs comprise a set of outputs, and said apparatus further

comprises means for selecting one of said outputs in response to at least one additional bit included in at least one of said plurality of inputs.

79. An article of manufacture according to claim 76, wherein the look-up table outputs corresponding to the plurality of inputs comprise a set of outputs, and said article further comprises computer readable code means which, when executed, will cause the step of selecting one of said outputs in response to at least one additional bit included in at least one of said plurality of inputs.

**EVIDENCE APPENDIX:**

There is no evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 or any other evidence entered by the Examiner and relied upon by Appellant in the appeal.

**RELATED PROCEEDINGS APPENDIX**

There are no decisions rendered by a court or the Board in any proceeding identified about in Section II pursuant to 37 C.F.R. § 41.37(c)(1)(ii).